DSC 465, Homework 1

Kefu Zhu

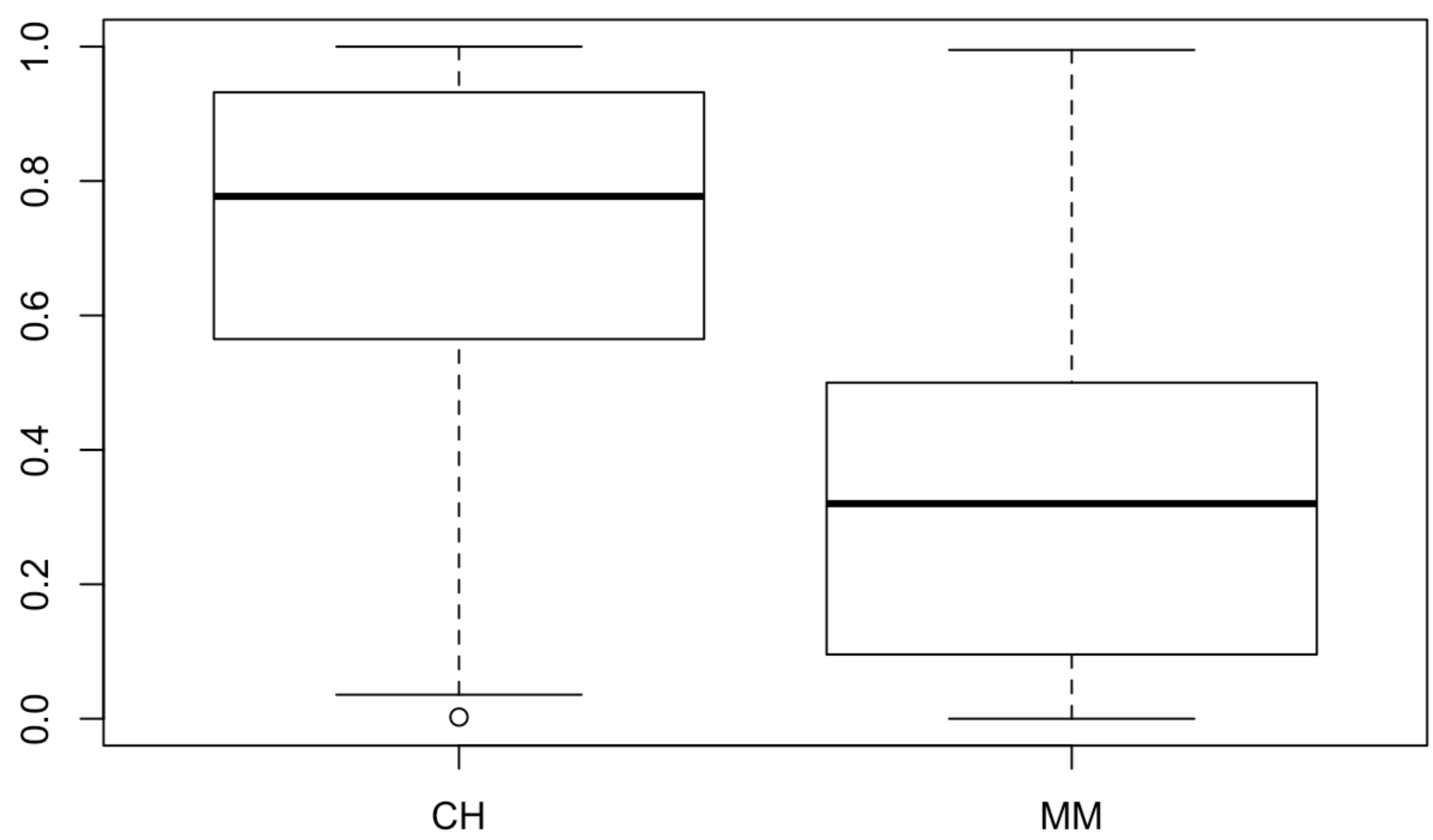
02/08/2019

Question 1

```r
# Load library
library(ISLR)
library(MASS)
```

(a)

```r
# Boxplot
boxplot(OJ$LoyalCH ~ OJ$Purchase)
```



From the above box plot, we can clearly see a difference between two purchase groups, indicating the `LoyalCH` score is different between these two groups.

```r
wilcox.test(OJ[OJ$Purchase == 'CH','LoyalCH'], OJ[OJ$Purchase == 'MM','LoyalCH'])
```

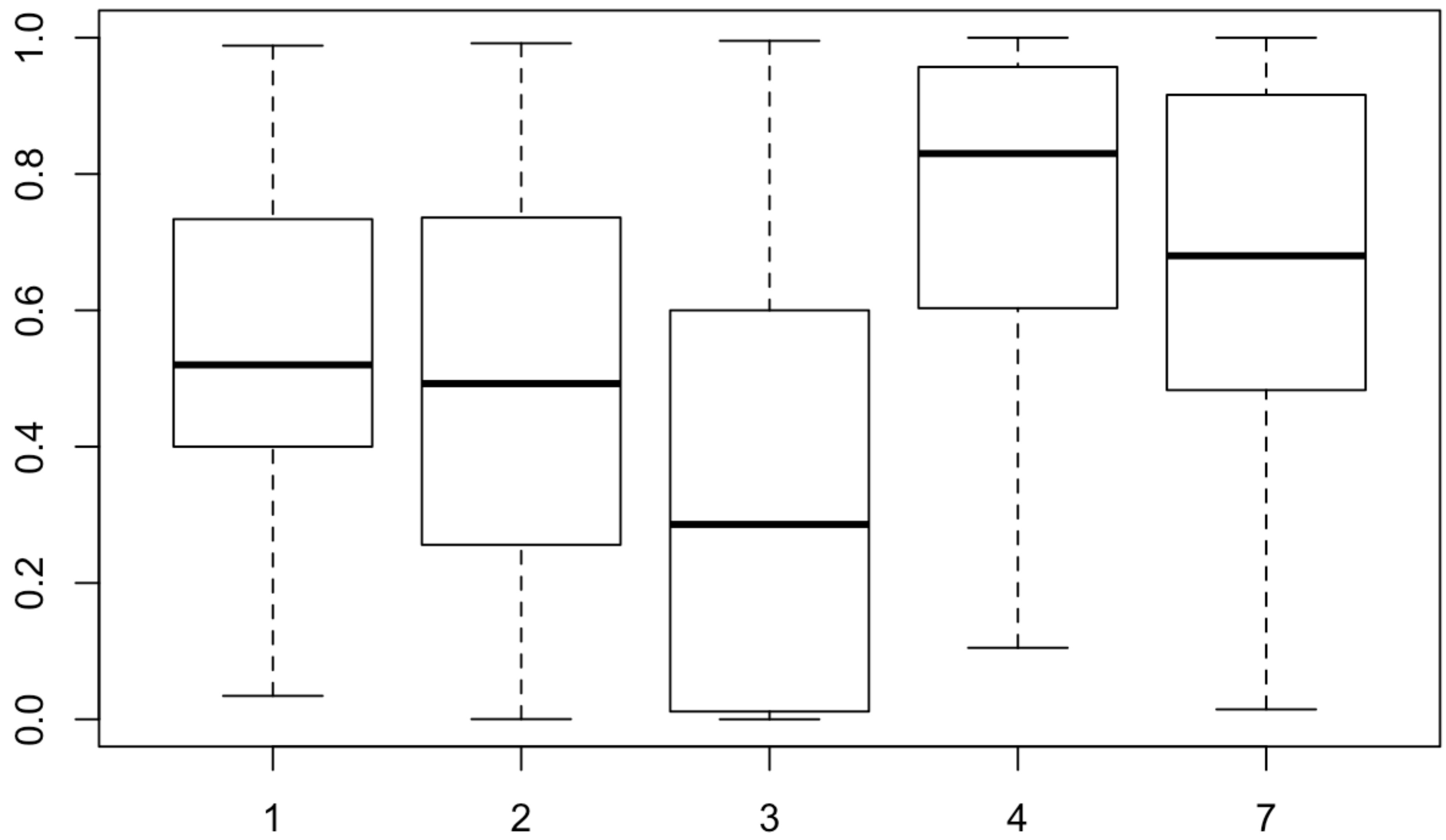
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  OJ[OJ$Purchase == "CH", "LoyalCH"] and OJ[OJ$Purchase == "MM", "LoyalCH"]
## W = 238000, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Answer:

Because the p-value from wilcoxon rank sum test is smaller than 0.05, we **reject the null hypothesis** that the median of `LoyalCH` score between two purchase groups (CH and MM) is the same.

(b)

```
# Boxplot
boxplot(OJ$LoyalCH ~ OJ$StoreID)
```




```
fit = aov(OJ$LoyalCH ~ as.factor(OJ$StoreID))
summary(fit)
```

```
##                         Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(OJ$StoreID)    4  18.94   4.734   61.21 <2e-16 ***
## Residuals             1065  82.37   0.077
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

Because the p-value from ANOVA is smaller than 0.05, we **reject the null hypothesis** that the mean loyalty score among stores are the same and conclude that mean loyalty score varies by store at a significance level of 5%.

(c)

```
fit.Tukey = TukeyHSD(fit)
fit.Tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = OJ$LoyalCH ~ as.factor(OJ$StoreID))
##
## $`as.factor(OJ$StoreID)`
##            diff         lwr         upr     p adj
## 2-1 -0.05302437 -0.13226671  0.02621798 0.3576340
## 3-1 -0.19716965 -0.27856011 -0.11577918 0.0000000
## 4-1  0.22021359  0.13171167  0.30871550 0.0000000
## 7-1  0.10989969  0.03709693  0.18270246 0.0003846
## 3-2 -0.14414528 -0.21862665 -0.06966391 0.0000015
## 4-2  0.27323795  0.19104516  0.35543075 0.0000000
## 7-2  0.16292406  0.09793706  0.22791106 0.0000000
## 4-3  0.41738323  0.33311750  0.50164896 0.0000000
## 7-3  0.30706934  0.23947964  0.37465904 0.0000000
## 7-4 -0.11031389 -0.18631750 -0.03431028 0.0007422
```

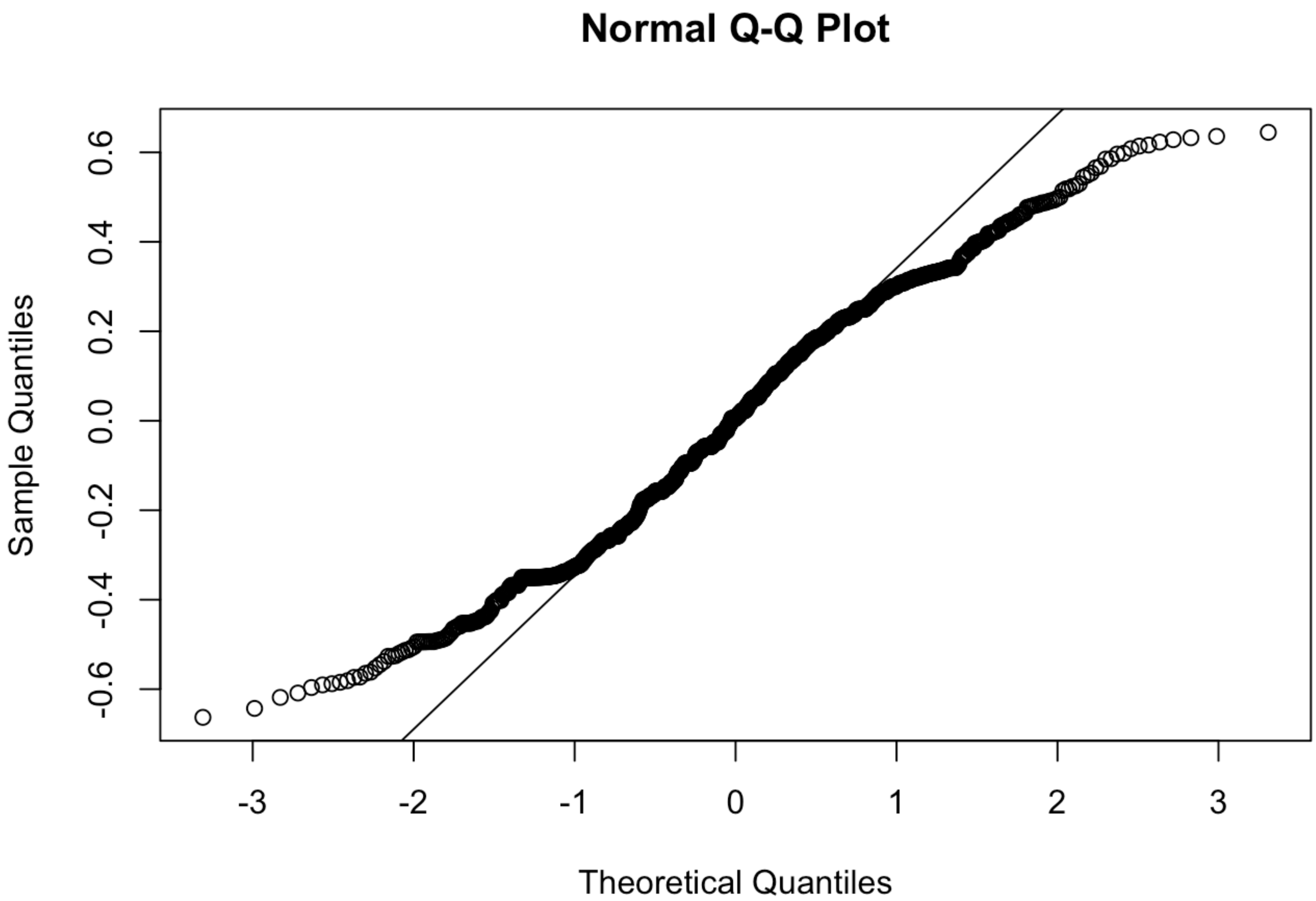
Answer:

Based on the result from Tukey's pairwise test, we can conclude the mean loyalty scores in different stores is in the following order, `Store 4 > Store 7 > Store 1 , Store 2 > Store 3` at a significance level of 5%.

Note: We cannot say whether mean loyalty score is higher or lower in `Store 1` or `Store 2`.

(d)

```
res = fit$residuals
qqnorm(res)
qqline(res)
```




```
# Empirical Rule
within_1sd = sum(abs(res - mean(res)) < sd(res))/length(res)
within_2sd = sum(abs(res - mean(res)) < 2*sd(res))/length(res)
within_3sd = sum(abs(res - mean(res)) < 3*sd(res))/length(res)
```

```
# Proportion of residuals within 1 standard deviation
within_1sd
```

```
## [1] 0.6130841
```

```
# Proportion of residuals within 2 standard deviation
within_2sd
```

```
## [1] 0.9747664
```

```
# Proportion of residuals within 3 standard deviation
within_3sd
```

```
## [1] 1
```

Answer:

Compare the actual proportion value with the empirical rule, 68%-95%-99.7%, we can clearly see the proportion is off by quite a lot. Therefore, we can conclude that the normality assumption is violated.

The reason we use the model residuals rather than the response variable is that the loyalty scores from different groups are assumed coming from indepent normal distributions rather than from one single normal distribution. So it is only reasonable for us to assess the normality of residuals within groups.

(e)

(i)

Define a random variable $M = F(x)$, for any $p \in [0, 1]$

$\because P(M \leq p) = P(F(x) \leq p) = P(x \leq F^{-1}(p)) = F(F^{-1}(p)) = p$

$\therefore F(x)$ follows uniform distribution

Define a random varialbe $N = 1 - F(x)$, for any $q \in [0, 1]$

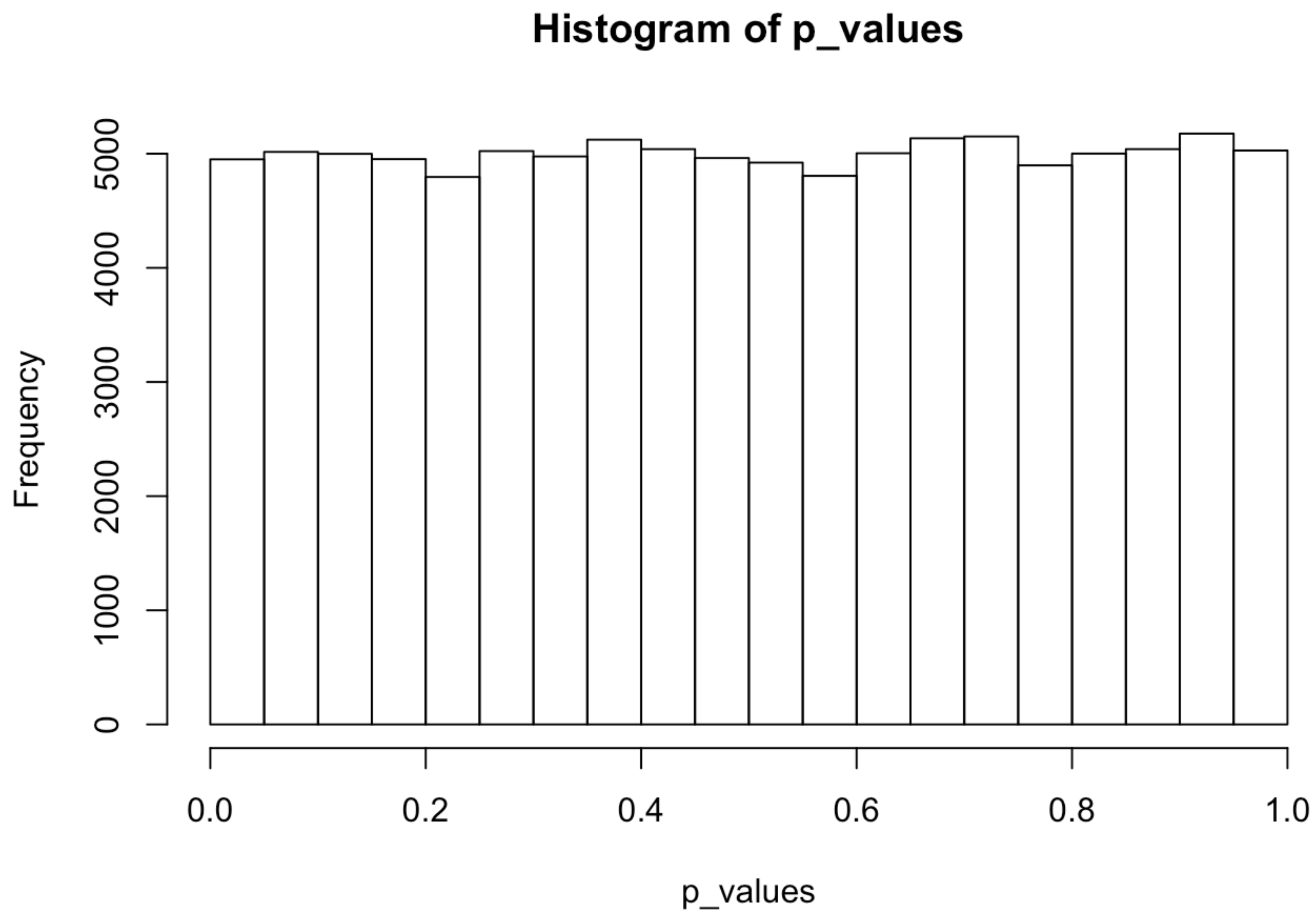
$\because P(N \leq q) = P(1 - F(x) \leq q) = P(1 - q \leq F(x)) = 1 - P(1 - q \geq F(x)) = 1 - (1 - q) = q$

$\therefore 1 - F(x)$ follows uniform distribution

(ii)

```
p_values <- c()
M = 100000
for(i in 1:M){
  seed = i*2
  set.seed(seed)
  boot_strap_index = sample(1:nrow(OJ), nrow(OJ), replace = T)
  anova.fit = aov(res[boot_strap_index] ~ as.factor(OJ[,'StoreID']))
  p_values = c(p_values, summary(anova.fit)[[1]][["Pr(>F)"]][1])
}
```

```
hist(p_values, nclass = 25)
```




```r
alpha_table = data.frame(matrix(ncol = 3, nrow = 4))
colnames(alpha_table) = c('alpha_hat','Z','SE')
index = 1

for(alpha in c(0.001,0.01,0.05,0.1)){
  alpha_hat = sum(p_values < alpha)/length(p_values)
  alpha_table[index,'alpha_hat'] = alpha_hat
  alpha_table[index,'SE'] = sqrt(alpha*(1-alpha)/M)
  alpha_table[index,'Z'] = (alpha_hat - alpha)/sqrt(alpha*(1-alpha)/M)
  index = index + 1
}
```

```r
alpha_table
```

```
##   alpha_hat          Z           SE
## 1   0.00103  0.3001501 9.994999e-05
## 2   0.01061  1.9387073 3.146427e-04
## 3   0.04951 -0.7109667 6.892024e-04
## 4   0.09967 -0.3478505 9.486833e-04
```

(iii)

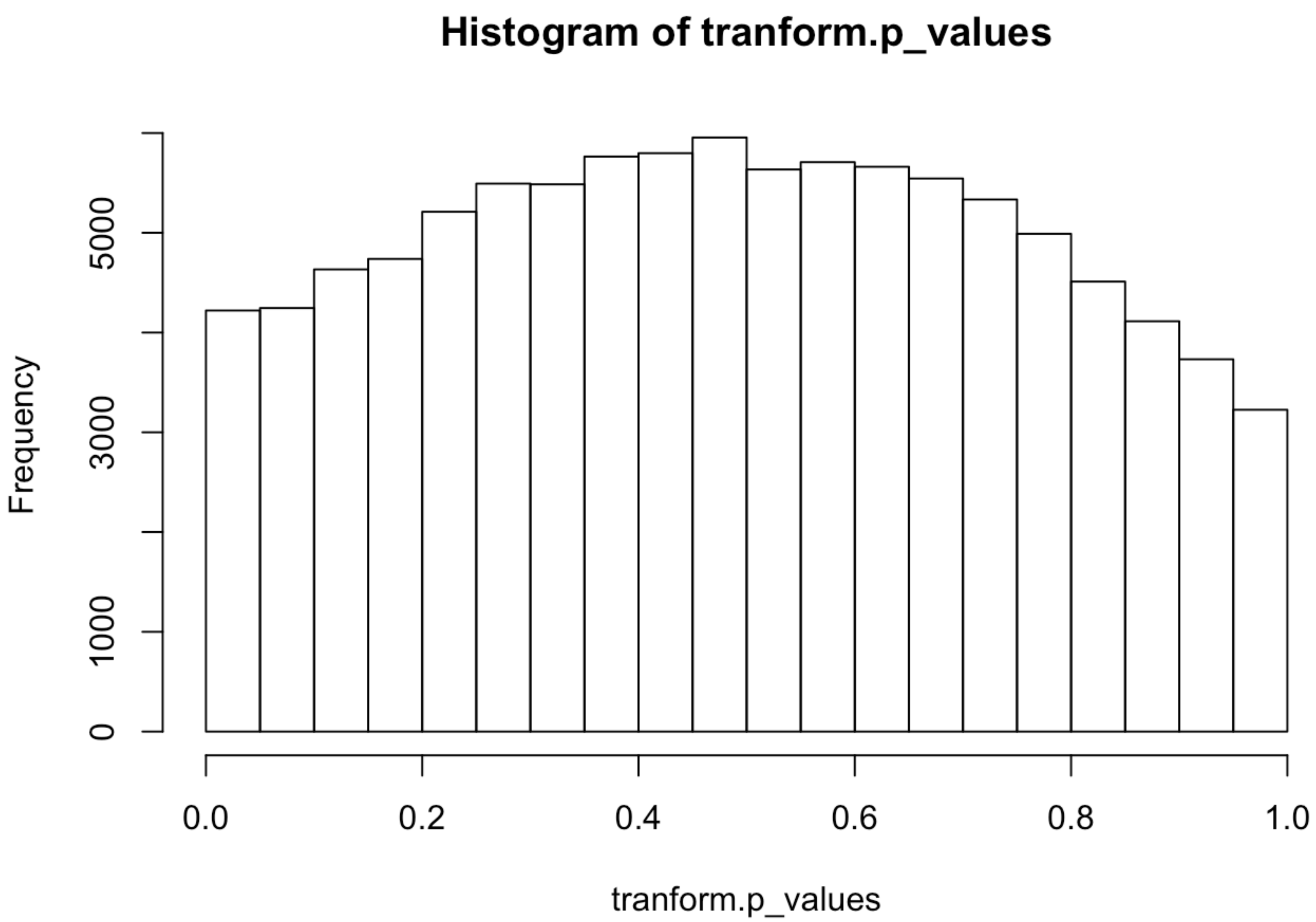
Answer: The standard error of $\hat{\alpha}$ for α = 0.1 is 9.48683310^{-4}. Given $M = 100000$, we can say that the bootstrap procedure is accurate in general.

(f)

```
y_star = 1/(1-OJ[OJ$StoreID == 7,'LoyalCH'])

tranform.p_values <- c()
M = 100000
for(i in 1:M){
  seed = i*2
  set.seed(seed)
  boot_strap_index = sample(length(y_star), nrow(OJ), replace = T)
  transform.anova.fit = aov(y_star[boot_strap_index] ~ as.factor(OJ[,'StoreID']))
  tranform.p_values = c(tranform.p_values, summary(transform.anova.fit)[[1]][["Pr(>F)"]][1])
}
```

```
hist(tranform.p_values, nclass = 25)
```

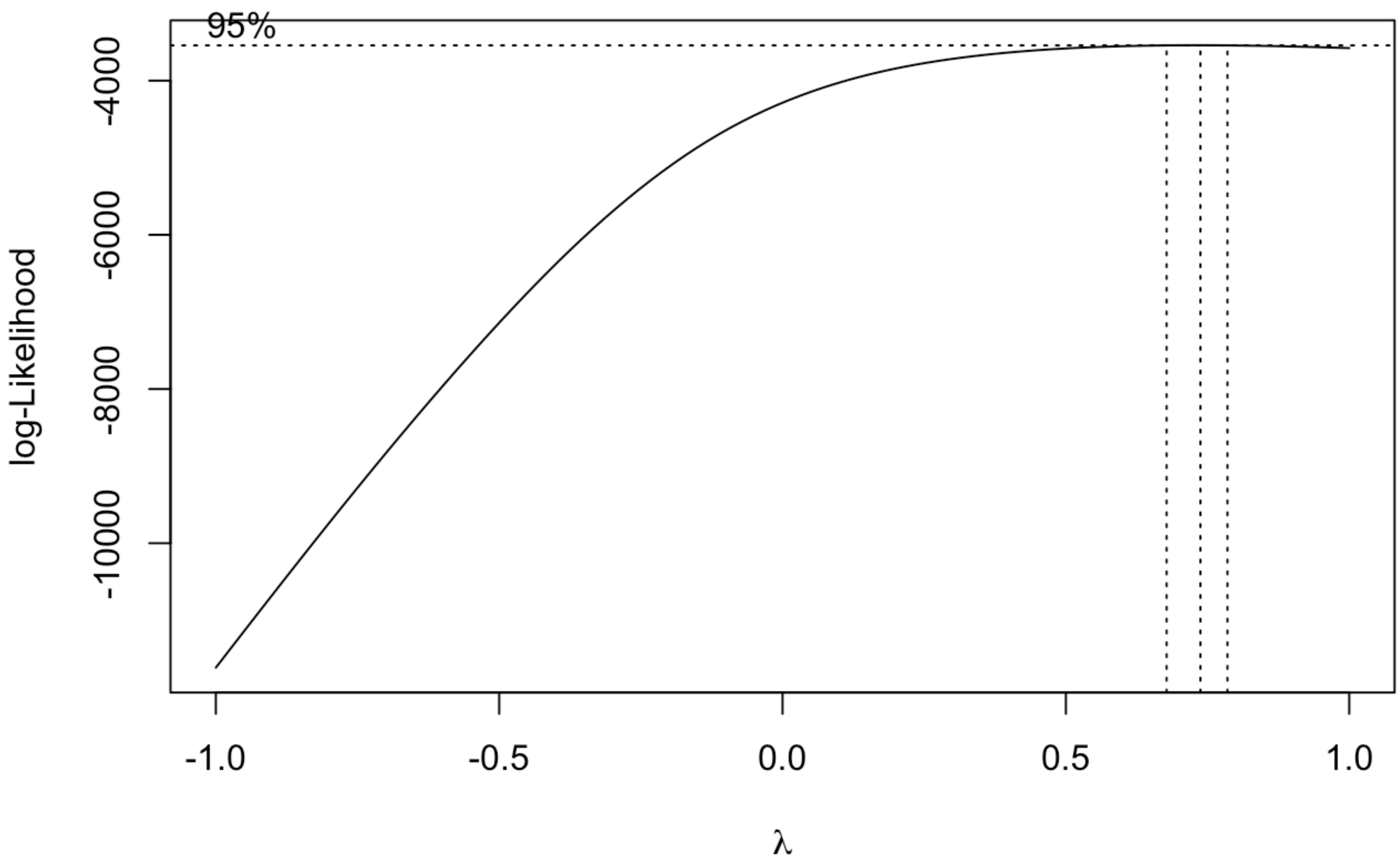


Answer:

Since the distribution of p-value is not uniform, it is not accurate

(g)

```
bc = boxcox(OJ$LoyalCH ~ OJ$StoreID, lambda = seq(-1, 1, length=10))
```




```
# Optimal lamda
best_lambda = bc$x[which.max(bc$y)]
best_lambda
```

```
## [1] 0.7373737
```

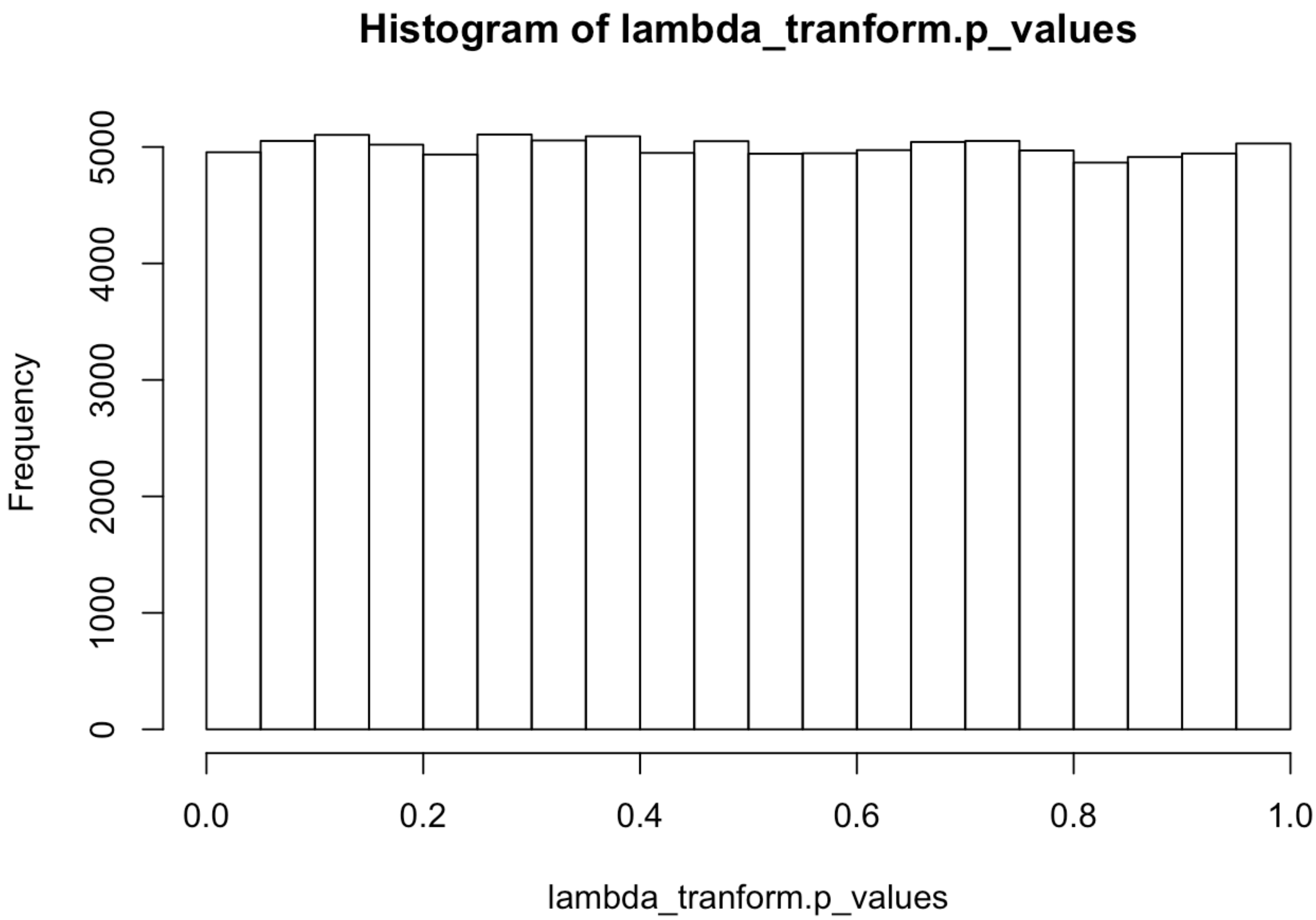
Based on the plot above, the optimal value for λ is `bc$x[which.max(bc$y)]`. Therefore we perform the following Box-Cox transformation

$$y^{\lambda} = \frac{y^{\lambda}-1}{\lambda}$$

```
y_lambda = (OJ[OJ$StoreID == 7,'LoyalCH']**best_lambda - 1)/best_lambda

lambda_tranform.p_values <- c()
M = 100000
for(i in 1:M){
  seed = i*2
  set.seed(seed)
  boot_strap_index = sample(length(y_lambda), nrow(OJ), replace = T)
  transform.anova.fit = aov(y_lambda[boot_strap_index] ~ as.factor(OJ[,'StoreID']))
  lambda_tranform.p_values = c(lambda_tranform.p_values, summary(transform.anova.fit)[[1]][["Pr(>F)
"]][1])
}
```

```
hist(lambda_tranform.p_values, nclass = 25)
```



Answer:

Since the distribution of p-value is uniform, it is accurate

Question 2

(a)

```
lm.PriceDiff = lm(LoyalCH ~ PriceDiff, data = OJ)
summary(lm.PriceDiff)
```

```
##
## Call:
## lm(formula = LoyalCH ~ PriceDiff, data = OJ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5947 -0.2273  0.0162  0.2729  0.5119
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54847    0.01064  51.544  < 2e-16 ***
## PriceDiff    0.11819    0.03450   3.426 0.000636 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3063 on 1068 degrees of freedom
## Multiple R-squared:  0.01087,    Adjusted R-squared:  0.009944
## F-statistic: 11.74 on 1 and 1068 DF,  p-value: 0.000636
```

Answer:

Based on the model summary, because both the p-value for `PriceDiff` and the p-value for the F-test is smaller than 0.05, we can conclude that `LoyalCH` varies with `PriceDiff` at a significance level of 5%.

(b)

```
lm.StoreID = lm(LoyalCH ~ as.factor(StoreID),data = OJ)
summary(lm.StoreID)
```

```
##
## Call:
## lm(formula = LoyalCH ~ as.factor(StoreID), data = OJ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6631 -0.2339  0.0053  0.2296  0.6448
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.54773    0.02220  24.678  < 2e-16 ***
## as.factor(StoreID)2 -0.05302    0.02900  -1.828   0.0678 .
## as.factor(StoreID)3 -0.19717    0.02979  -6.619 5.70e-11 ***
## as.factor(StoreID)4  0.22021    0.03239   6.799 1.75e-11 ***
## as.factor(StoreID)7  0.10990    0.02664   4.125 4.00e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2781 on 1065 degrees of freedom
## Multiple R-squared:  0.1869, Adjusted R-squared:  0.1839
## F-statistic: 61.21 on 4 and 1065 DF,  p-value: < 2.2e-16
```

Answer:

The smaller than 0.05 p-value of F-test from the model summary indicates that `LoyalCH` varies with `StoreID`, which also verifies the result from **Q1 (b)** saying the mean loyalty score among stores are NOT the same.

(c)

```
lm.PriceDiff_StoreID = lm(LoyalCH ~ PriceDiff + as.factor(StoreID), data = OJ)
summary(lm.PriceDiff_StoreID)
```

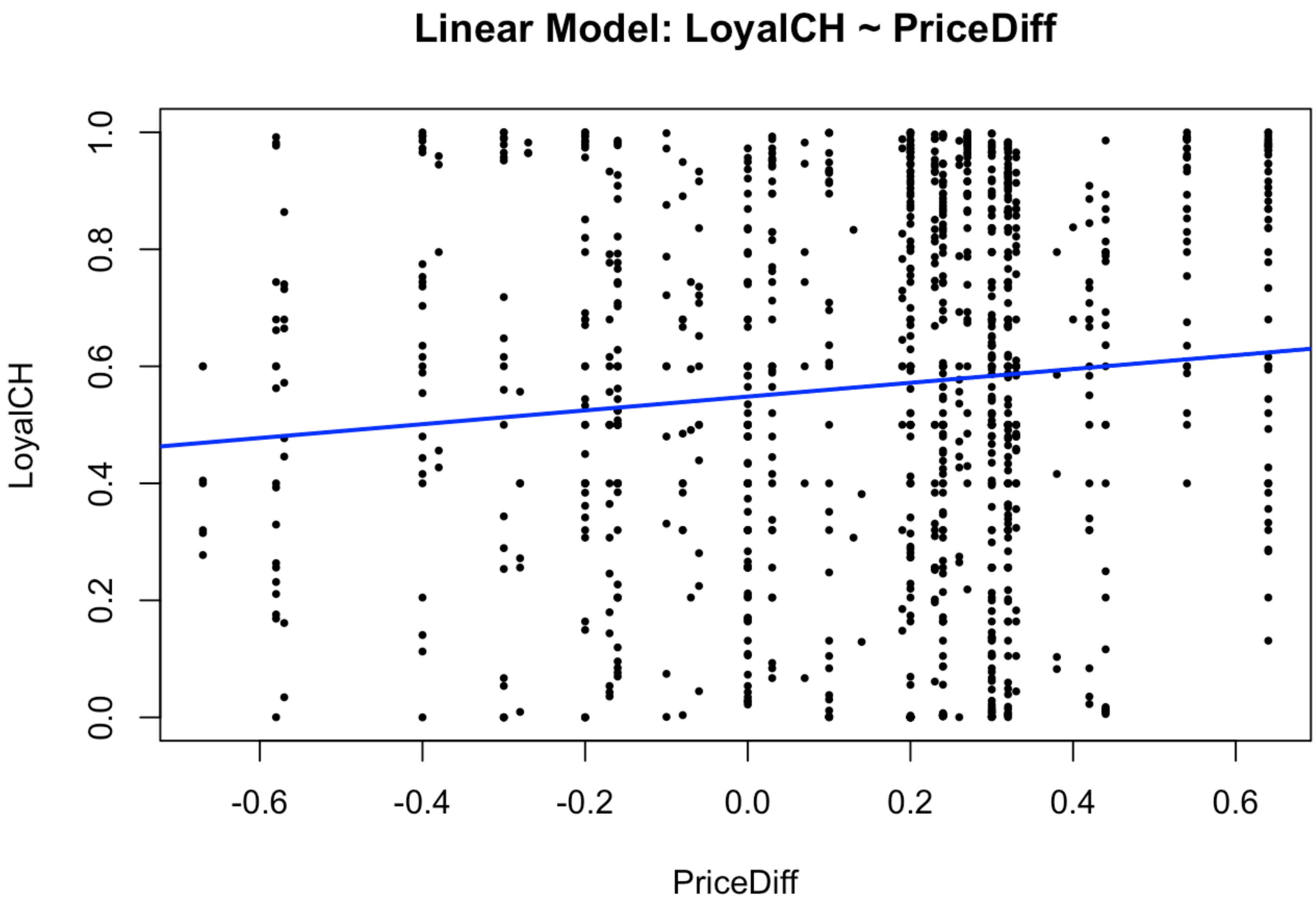
```
##
## Call:
## lm(formula = LoyalCH ~ PriceDiff + as.factor(StoreID), data = OJ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66280 -0.22003  0.00733  0.22806  0.67917
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.53980    0.02220  24.315  < 2e-16 ***
## PriceDiff             0.10990    0.03176   3.460 0.000561 ***
## as.factor(StoreID)2 -0.05714    0.02888  -1.979 0.048082 *
## as.factor(StoreID)3 -0.20711    0.02977  -6.956 6.09e-12 ***
## as.factor(StoreID)4  0.21686    0.03224   6.727 2.82e-11 ***
## as.factor(StoreID)7  0.09468    0.02687   3.524 0.000444 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2767 on 1064 degrees of freedom
## Multiple R-squared:  0.196,  Adjusted R-squared:  0.1922
## F-statistic: 51.86 on 5 and 1064 DF,  p-value: < 2.2e-16
```

```
lm.PriceDiff_StoreID_interactions = lm(LoyalCH ~ PriceDiff * as.factor(StoreID), data = OJ)
summary(lm.PriceDiff_StoreID_interactions)
```

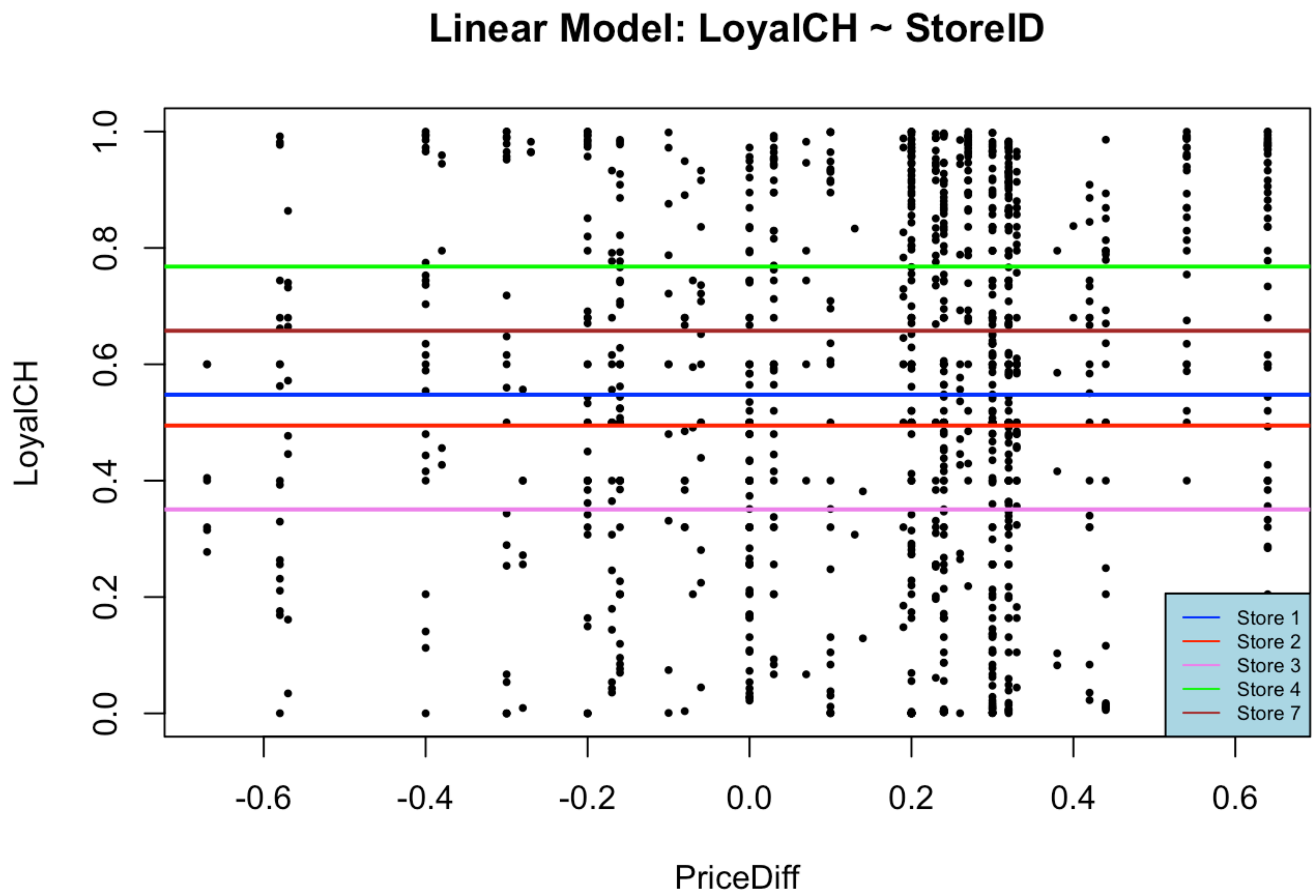
```
##
## Call:
## lm(formula = LoyalCH ~ PriceDiff * as.factor(StoreID), data = OJ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66389 -0.22608  0.00451  0.22162  0.64480
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.53501    0.02265  23.618  < 2e-16 ***
## PriceDiff                     0.17639    0.07261   2.429   0.0153 *
## as.factor(StoreID)2          -0.04131    0.03019  -1.369   0.1714
## as.factor(StoreID)3          -0.18456    0.03391  -5.443 6.51e-08 ***
## as.factor(StoreID)4           0.23402    0.03441   6.801 1.73e-11 ***
## as.factor(StoreID)7           0.08254    0.02899   2.847   0.0045 **
## PriceDiff:as.factor(StoreID)2 -0.16721    0.09909  -1.687   0.0918 .
## PriceDiff:as.factor(StoreID)3 -0.17571    0.12101  -1.452   0.1468
## PriceDiff:as.factor(StoreID)4 -0.18698    0.12989  -1.439   0.1503
## PriceDiff:as.factor(StoreID)7  0.01395    0.08842   0.158   0.8747
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2761 on 1060 degrees of freedom
## Multiple R-squared:  0.2021, Adjusted R-squared:  0.1953
## F-statistic: 29.83 on 9 and 1060 DF,  p-value: < 2.2e-16
```

```
OJ_Store1 = OJ[OJ$StoreID == 1,]
OJ_Store2 = OJ[OJ$StoreID == 2,]
OJ_Store3 = OJ[OJ$StoreID == 3,]
OJ_Store4 = OJ[OJ$StoreID == 4,]
OJ_Store7 = OJ[OJ$StoreID == 7,]
```

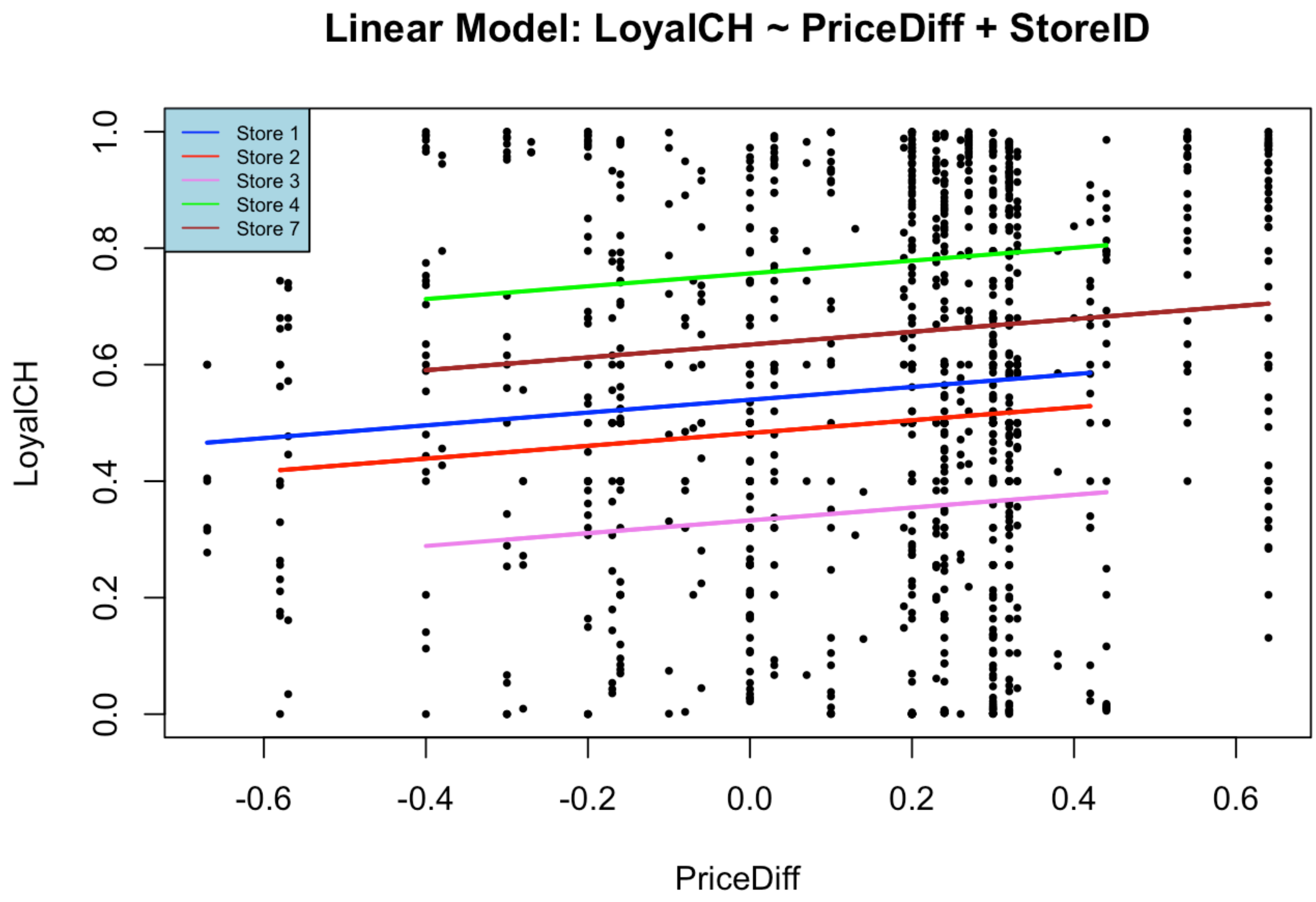
```
plot(OJ$LoyalCH ~ OJ$PriceDiff, pch = 20, cex = 0.6,
    xlab = 'PriceDiff',
    ylab = 'LoyalCH',
    main = 'Linear Model: LoyalCH ~ PriceDiff')
abline(lm.PriceDiff, lwd = 2, col = 'blue')
```



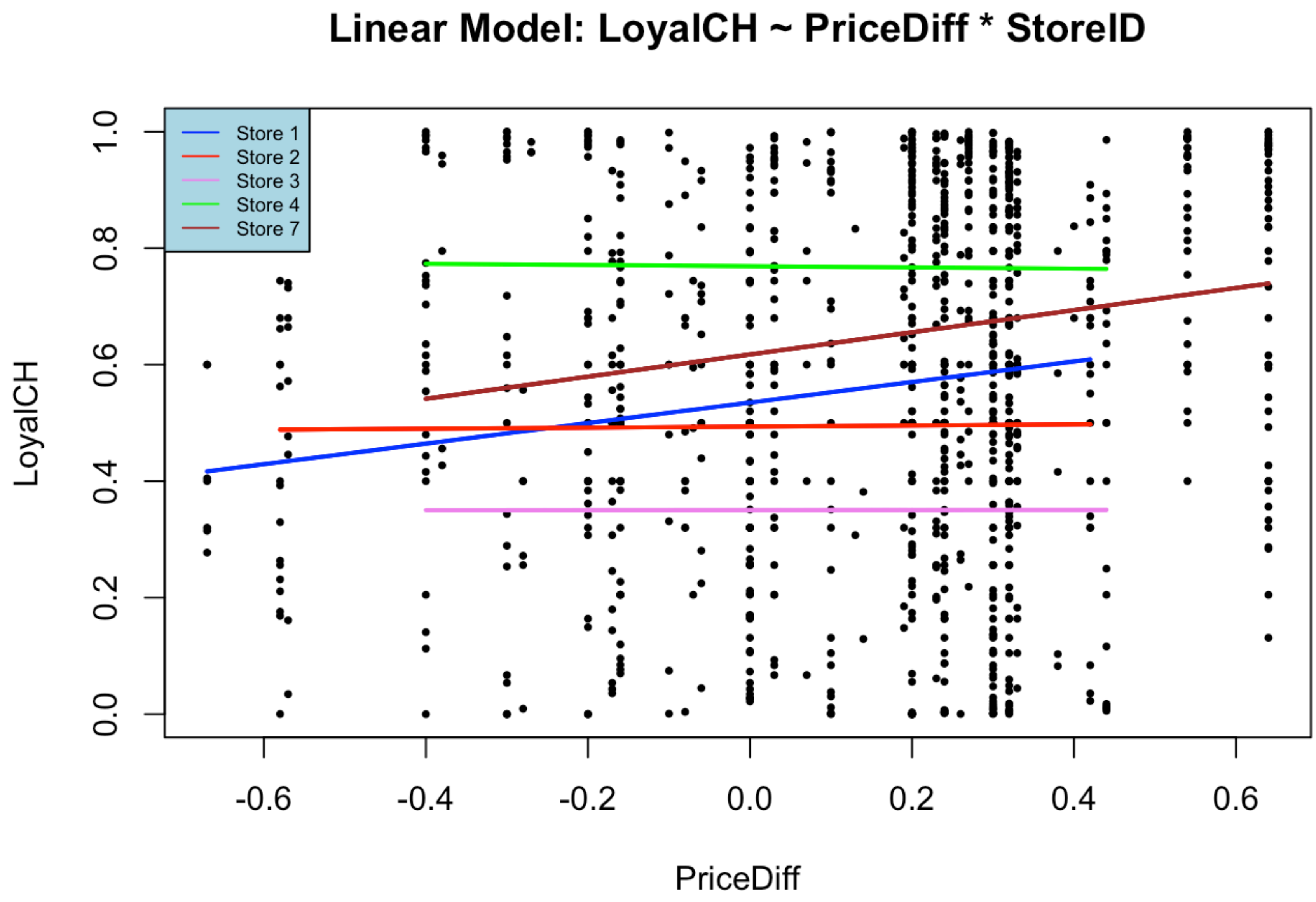

```
plot(OJ$LoyalCH ~ OJ$PriceDiff, pch = 20, cex = 0.6,
    xlab = 'PriceDiff',
    ylab = 'LoyalCH',
    main = 'Linear Model: LoyalCH ~ StoreID')
abline(h = lm.StoreID$coefficients[1],
       lwd = 2, col = 'blue')
abline(h = lm.StoreID$coefficients[1] + lm.StoreID$coefficients[2],
       lwd = 2, col = 'red')
abline(h = lm.StoreID$coefficients[1] + lm.StoreID$coefficients[3],
       lwd = 2, col = 'violet')
abline(h = lm.StoreID$coefficients[1] + lm.StoreID$coefficients[4],
       lwd = 2, col = 'green')
abline(h = lm.StoreID$coefficients[1] + lm.StoreID$coefficients[5],
       lwd = 2, col = 'brown')
legend('bottomright', cex = 0.6,
       legend = c('Store 1','Store 2','Store 3','Store 4','Store 7'),
       col = c('blue','red','violet','green','brown'),
       lty = 1,
       bg = 'lightblue')
```




```
plot(OJ$LoyalCH ~ OJ$PriceDiff, pch = 20, cex = 0.6,
    xlab = 'PriceDiff',
    ylab = 'LoyalCH',
    main = 'Linear Model: LoyalCH ~ PriceDiff + StoreID')
Store_1_predict = predict(lm.PriceDiff_StoreID, newdata = OJ_Store1)
Store_2_predict = predict(lm.PriceDiff_StoreID, newdata = OJ_Store2)
Store_3_predict = predict(lm.PriceDiff_StoreID, newdata = OJ_Store3)
Store_4_predict = predict(lm.PriceDiff_StoreID, newdata = OJ_Store4)
Store_7_predict = predict(lm.PriceDiff_StoreID, newdata = OJ_Store7)
lines(x = OJ_Store1$PriceDiff, y = Store_1_predict, lwd = 2, col = 'blue')
lines(x = OJ_Store2$PriceDiff, y = Store_2_predict, lwd = 2, col = 'red')
lines(x = OJ_Store3$PriceDiff, y = Store_3_predict, lwd = 2, col = 'violet')
lines(x = OJ_Store4$PriceDiff, y = Store_4_predict, lwd = 2, col = 'green')
lines(x = OJ_Store7$PriceDiff, y = Store_7_predict, lwd = 2, col = 'brown')
legend('topleft', cex = 0.6,
       legend = c('Store 1','Store 2','Store 3','Store 4','Store 7'),
       col = c('blue','red','violet','green','brown'),
       lty = 1,
       bg = 'lightblue')
```




```
plot(OJ$LoyalCH ~ OJ$PriceDiff, pch = 20, cex = 0.6,
    xlab = 'PriceDiff',
    ylab = 'LoyalCH',
    main = 'Linear Model: LoyalCH ~ PriceDiff * StoreID')
Store_1_predict = predict(lm.PriceDiff_StoreID_interactions, newdata = OJ_Store1)
Store_2_predict = predict(lm.PriceDiff_StoreID_interactions, newdata = OJ_Store2)
Store_3_predict = predict(lm.PriceDiff_StoreID_interactions, newdata = OJ_Store3)
Store_4_predict = predict(lm.PriceDiff_StoreID_interactions, newdata = OJ_Store4)
Store_7_predict = predict(lm.PriceDiff_StoreID_interactions, newdata = OJ_Store7)
lines(x = OJ_Store1$PriceDiff, y = Store_1_predict, lwd = 2, col = 'blue')
lines(x = OJ_Store2$PriceDiff, y = Store_2_predict, lwd = 2, col = 'red')
lines(x = OJ_Store3$PriceDiff, y = Store_3_predict, lwd = 2, col = 'violet')
lines(x = OJ_Store4$PriceDiff, y = Store_4_predict, lwd = 2, col = 'green')
lines(x = OJ_Store7$PriceDiff, y = Store_7_predict, lwd = 2, col = 'brown')
legend('topleft', cex = 0.6,
       legend = c('Store 1','Store 2','Store 3','Store 4','Store 7'),
       col = c('blue','red','violet','green','brown'),
       lty = 1,
       bg = 'lightblue')
```



(d)

```
SSE_table = data.frame(matrix(ncol = 3, nrow = 4))
colnames(SSE_table) = c('Model','SSE','df')
SSE_table$Model = seq(1,4)
SSE_table$SSE = c(sum(lm.PriceDiff$residuals**2),
                  sum(lm.StoreID$residuals**2),
                  sum(lm.PriceDiff_StoreID$residuals**2),
                  sum(lm.PriceDiff_StoreID_interactions$residuals**2))
SSE_table$df = c(lm.PriceDiff$df.residual,
                 lm.StoreID$df.residual,
                 lm.PriceDiff_StoreID$df.residual,
                 lm.PriceDiff_StoreID_interactions$df.residual)
```

```
knitr::kable(SSE_table)
```

Model	SSE	df
1	100.20472	1068
2	82.37044	1065
3	81.45389	1064
4	80.83364	1060

- Both Model 1 and 2 are a reduced models from Model 3 or Model 4
- Model 3 is a reduced model from Model 4

```
# SSE
SSE_m1 = SSE_table[SSE_table$Model == 1,'SSE']
SSE_m2 = SSE_table[SSE_table$Model == 2,'SSE']
SSE_m3 = SSE_table[SSE_table$Model == 3,'SSE']
SSE_m4 = SSE_table[SSE_table$Model == 4,'SSE']
# Number of parameters (excluding the intercept)
df_m1 = nrow(OJ) - SSE_table[SSE_table$Model == 1,'df'] - 1
df_m2 = nrow(OJ) - SSE_table[SSE_table$Model == 2,'df'] - 1
df_m3 = nrow(OJ) - SSE_table[SSE_table$Model == 3,'df'] - 1
df_m4 = nrow(OJ) - SSE_table[SSE_table$Model == 4,'df'] - 1
# F-test (F_full.model_reduced.model)
F_3_1 = ((SSE_m1 - SSE_m3)/(df_m3 - df_m1))/(SSE_m3/(nrow(OJ) - (df_m3 + 1)))
F_3_2 = ((SSE_m2 - SSE_m3)/(df_m3 - df_m2))/(SSE_m3/(nrow(OJ) - (df_m3 + 1)))
F_4_3 = ((SSE_m3 - SSE_m4)/(df_m4 - df_m3))/(SSE_m4/(nrow(OJ) - (df_m4 + 1)))
```

Goodness of fit test

- **Model 3 (full) vs. Model 1 (reduced)**:

  F test statistic is 61.2336728, with degree of freedom of 4 and 1064.

  Because the F statistic is greater than the critical value $\chi^2_{4,\infty,0.05} = 2.37$, so we can **REJECT** the null hypothesis that the reduced model is beteer at a significance level of 5%, andconclude that **Model 3 improves Model 1**.

- **Model 3 (full) vs. Model 2 (reduced)**:

  F test statistic is 11.9724792, with degree of freedom of 1 and 1064.

  Because the F statistic is greater than the critical value $\chi^2_{1,\infty,0.05} = 3.84$, so we can **REJECT** the null hypothesis that the reduced model is beteer at a significance level of 5%, and conclude that **Model 3 improves Model 2**.

- **Model 4 (full) vs. Model 3 (reduced)**:

  F test statistic is 2.0333783, with degree of freedom of 4 and 1060.

  Because the F statistic is smaller than the critical value $\chi^2_{4,\infty,0.05} = 2.37$, so we **FAIL TO REJECT** the null hypothesis that the reduced model is beteer at a significance level of 5%, and conclude that **Model 4 DOES NOT significantly improve Model 3**.

```
anova(lm.PriceDiff, lm.PriceDiff_StoreID)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ PriceDiff
## Model 2: LoyalCH ~ PriceDiff + as.factor(StoreID)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1   1068 100.205
## 2   1064  81.454  4    18.751 61.234 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm.StoreID, lm.PriceDiff_StoreID)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ as.factor(StoreID)
## Model 2: LoyalCH ~ PriceDiff + as.factor(StoreID)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1065 82.370
## 2   1064 81.454  1   0.91655 11.973 0.0005613 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm.PriceDiff_StoreID, lm.PriceDiff_StoreID_interactions)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ PriceDiff + as.factor(StoreID)
## Model 2: LoyalCH ~ PriceDiff * as.factor(StoreID)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1   1064 81.454
## 2   1060 80.834  4   0.62025 2.0334 0.08763 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: Results from `anova` function verify that previous calculations are correct.

(e)

```
m2_Store1 = lm(LoyalCH ~ as.factor(StoreID) + I(PriceDiff*(StoreID == 1)), data = OJ)
m2_Store2 = lm(LoyalCH ~ as.factor(StoreID) + I(PriceDiff*(StoreID == 2)), data = OJ)
m2_Store3 = lm(LoyalCH ~ as.factor(StoreID) + I(PriceDiff*(StoreID == 3)), data = OJ)
m2_Store4 = lm(LoyalCH ~ as.factor(StoreID) + I(PriceDiff*(StoreID == 4)), data = OJ)
m2_Store7 = lm(LoyalCH ~ as.factor(StoreID) + I(PriceDiff*(StoreID == 7)), data = OJ)
```

```
anova(lm.StoreID,m2_Store1)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ as.factor(StoreID)
## Model 2: LoyalCH ~ as.factor(StoreID) + I(PriceDiff * (StoreID == 1))
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1   1065 82.37
## 2   1064 81.92  1   0.45005 5.8454 0.01579 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm.StoreID,m2_Store2)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ as.factor(StoreID)
## Model 2: LoyalCH ~ as.factor(StoreID) + I(PriceDiff * (StoreID == 2))
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1065 82.370
## 2   1064 82.369  1  0.001413 0.0183 0.8926
```

```
anova(lm.StoreID,m2_Store3)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ as.factor(StoreID)
## Model 2: LoyalCH ~ as.factor(StoreID) + I(PriceDiff * (StoreID == 3))
##   Res.Df   RSS Df  Sum of Sq  F Pr(>F)
## 1   1065 82.37
## 2   1064 82.37  1 3.7098e-06  0 0.9945
```

```
anova(lm.StoreID,m2_Store4)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ as.factor(StoreID)
## Model 2: LoyalCH ~ as.factor(StoreID) + I(PriceDiff * (StoreID == 4))
##   Res.Df   RSS Df  Sum of Sq      F Pr(>F)
## 1   1065 82.37
## 2   1064 82.37  1 0.00073793 0.0095 0.9222
```

```
anova(lm.StoreID,m2_Store7)
```

```
## Analysis of Variance Table
##
## Model 1: LoyalCH ~ as.factor(StoreID)
## Model 2: LoyalCH ~ as.factor(StoreID) + I(PriceDiff * (StoreID == 7))
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1065 82.370
## 2   1064 81.286  1    1.0846 14.197 0.0001737 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

- **Model 2 +** $\beta' \times I(PriceDiff * (StoreID == 1))$ **(full) vs. Model 2 (reduced)**: The p-value from F-test is **0.01579**
- **Model 2 +** $\beta' \times I(PriceDiff * (StoreID == 2))$ **(full) vs. Model 2 (reduced)**: The p-value from F-test is **0.8926**
- **Model 2 +** $\beta' \times I(PriceDiff * (StoreID == 3))$ **(full) vs. Model 2 (reduced)**: The p-value from F-test is **0.9945**
- **Model 2 +** $\beta' \times I(PriceDiff * (StoreID == 4))$ **(full) vs. Model 2 (reduced)**: The p-value from F-test is **0.9222**
- **Model 2 +** $\beta' \times I(PriceDiff * (StoreID == 7))$ **(full) vs. Model 2 (reduced)**: The p-value from F-test is **0.0001737**

(f)

Answer:

Based on the Beonferroni multiple test procedure, since we want to simultaneously report 5 confidence intervals with $\alpha_{FWE} = 0.05$, we need to compare the F test statistics with $\chi^2_{1,\infty,0.01}$

Because $\chi^2_{1,\infty,0.01} = 6.63$, only the F test statistic, 14.197, from the F-test on

Model 2 (full) vs. Model 2 + $\beta' \times I(PriceDiff * (StoreID == 7))$

is larger than the critical value. Therefore we can conclude that `LoyalCH` varies with `PriceDiff` in `Store 7` at a significance level of 5%.

Question 3

(a)

$$\because \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_q \end{bmatrix}, X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1q} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2q} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nq} \end{bmatrix}$$

$$\therefore X^TX = \begin{bmatrix} x_{11} & x_{21} & x_{31} & \dots & x_{q1} \\ x_{12} & x_{22} & x_{32} & \dots & x_{q2} \\ \dots & \dots & \dots & \dots & \dots \\ x_{1n} & x_{2n} & x_{3n} & \dots & x_{qn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1q} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2q} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nq} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_{i1}^2 & \sum_{i=1}^n x_{i1}x_{i2} & \dots & \sum_{i=1}^n x_{i1}x_{iq} \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^n x_{iq}x_{i1} & \sum_{i=1}^n x_{iq}x_{i2} & \dots & \sum_{i=1}^n x_{iq}^2 \end{bmatrix}$$

Then we can observe that X^TX is a diagonal matrix if and only if $\sum_{i=1}^n x_{ij}x_{ik} = 0$ for each pair $j \neq k$

Therefore, we can see $\sum_{\hat{\beta}}$ is also a diagonal matrix $\Rightarrow$ the covariance of $\hat{\beta}$ is zero $\Rightarrow$ regression coefficients are mutually uncorrelated

(b)

$\because \hat{\beta} = (X^TX)^{-1}X^Ty, \hat{\beta'} = (X'^TX)^{-1}X'^Ty, z_i = x_iA$

To avoid collinearity, we substitude X' with XA

$\therefore \hat{\beta'} = ((XA)^TXA)^{-1}(XA)^Ty$

For $y_i = x_i\beta, \hat{y_i} = x_i(X^TX)^{-1}X^Ty$

For $y_i = z_i\beta', \hat{y_i'} = x_iA((XA)^TXA)^{-1}(XA)^Ty = x_iA(X^TA^TXA)^{-1}X^TA^T$

Because A is a matrix of $(p+1) \times (p+1)$, $(X^TA^TXA)^{-1} = A^{-1}(X^TX)^{-1}(A^T)^{-1} \to \hat{y_i'} = x_i(X^TX)^{-1}X^Ty$

Therefore, the two models are equivalent in the sense that the fitted values $\hat{y_i}$ must be the same.

In addition, we can derive the relationship between $\hat{\beta}$ and $\hat{\beta'}$

$$\begin{cases} \hat{\beta} = (X^TX)^{-1}X^Ty = X^{-1}y \\ \hat{\beta'} = ((XA)^TXA)^{-1}(XA)^Ty = A^{-1}X^{-1}y \end{cases} \Rightarrow \hat{\beta'} = A^{-1}\hat{\beta}$$

(c)

From part(a), we know that $X^T X$ is a diagonal matrix if and only if $\sum_{i=1}^n x_{ij}x_{ik} = 0$ for each pair $j \neq k$.

First we will try to prove the components of $\hat{\beta}'$ is uncorrelated:

From part (b), we substitude X' with XA. Then we have $X'^T X' = A^T X^T AX$

$$\because XA = \begin{bmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} 1 & -\bar{x} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & x_1 - \bar{x} \\ \dots & \dots \\ 1 & x_n - \bar{x} \end{bmatrix}$$

$$\therefore A^T X^T AX = \begin{bmatrix} 1 & \dots & 1 \\ x_1 - \bar{x} & \dots & x_n - \bar{x} \end{bmatrix} \begin{bmatrix} 1 & x_1 - \bar{x} \\ \dots & \dots \\ 1 & x_n - \bar{x} \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^n x_i - n\bar{x} \\ \sum_{i=1}^n x_i - n\bar{x} & \sum_{i=1}^n (x_i - \bar{x})^2 \end{bmatrix} = \begin{bmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x})^2 \end{bmatrix}$$

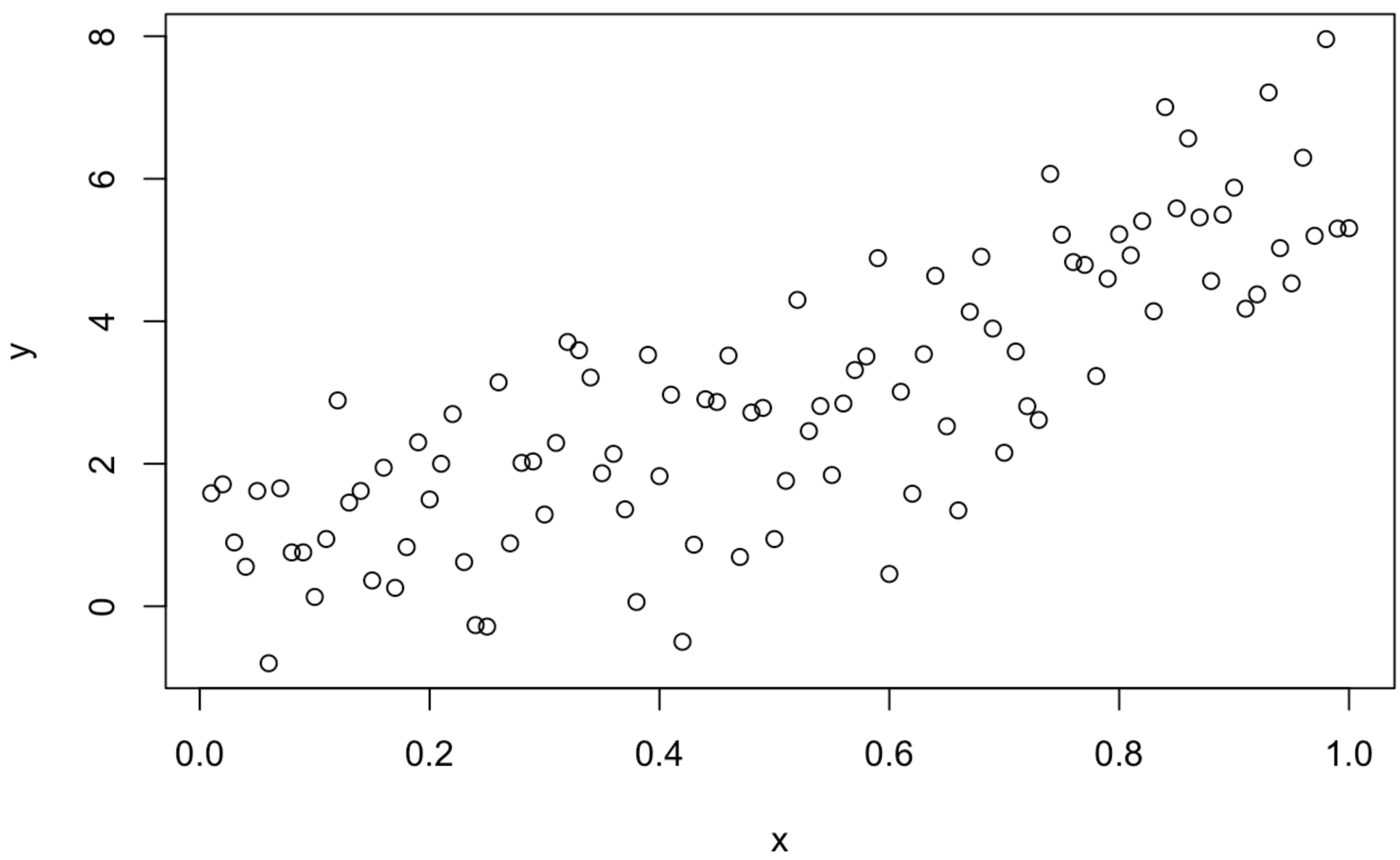
We prove the components of $\hat{\beta}'$ is uncorrelated.

Because from part(c), we know $\hat{\beta}' = A^{-1}\hat{\beta}$, so we also prove the components of $\hat{\beta}$ is uncorrelated.

(d)

(i)

```
set.seed(12345)
x = (1:100)/100
y = rnorm(100,mean=1+5*x^2,sd=1)
plot(x,y)
```



The model is $y = 5x^2 + \epsilon$, where $\epsilon \sim N(1,1)$

(ii)

```
fit1 = lm(y~poly(x,3,raw=T))
fit2 = lm(y~poly(x,3,raw=F))
```

```
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 3, raw = T))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6616 -0.8362  0.2023  0.6317  2.1481
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.0784     0.4690   2.299   0.0236 *
## poly(x, 3, raw = T)1   0.2263     4.0014   0.057   0.9550
## poly(x, 3, raw = T)2   5.7621     9.1813   0.628   0.5318
## poly(x, 3, raw = T)3  -0.8052     5.9772  -0.135   0.8931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.129 on 96 degrees of freedom
## Multiple R-squared:  0.6682, Adjusted R-squared:  0.6578
## F-statistic: 64.45 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 3, raw = F))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6616 -0.8362  0.2023  0.6317  2.1481
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.9369     0.1129  26.018  < 2e-16 ***
## poly(x, 3, raw = F)1  15.3256     1.1288  13.577  < 2e-16 ***
## poly(x, 3, raw = F)2   3.3847     1.1288   2.998  0.00345 **
## poly(x, 3, raw = F)3  -0.1521     1.1288  -0.135  0.89312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.129 on 96 degrees of freedom
## Multiple R-squared:  0.6682, Adjusted R-squared:  0.6578
## F-statistic: 64.45 on 3 and 96 DF,  p-value: < 2.2e-16
```

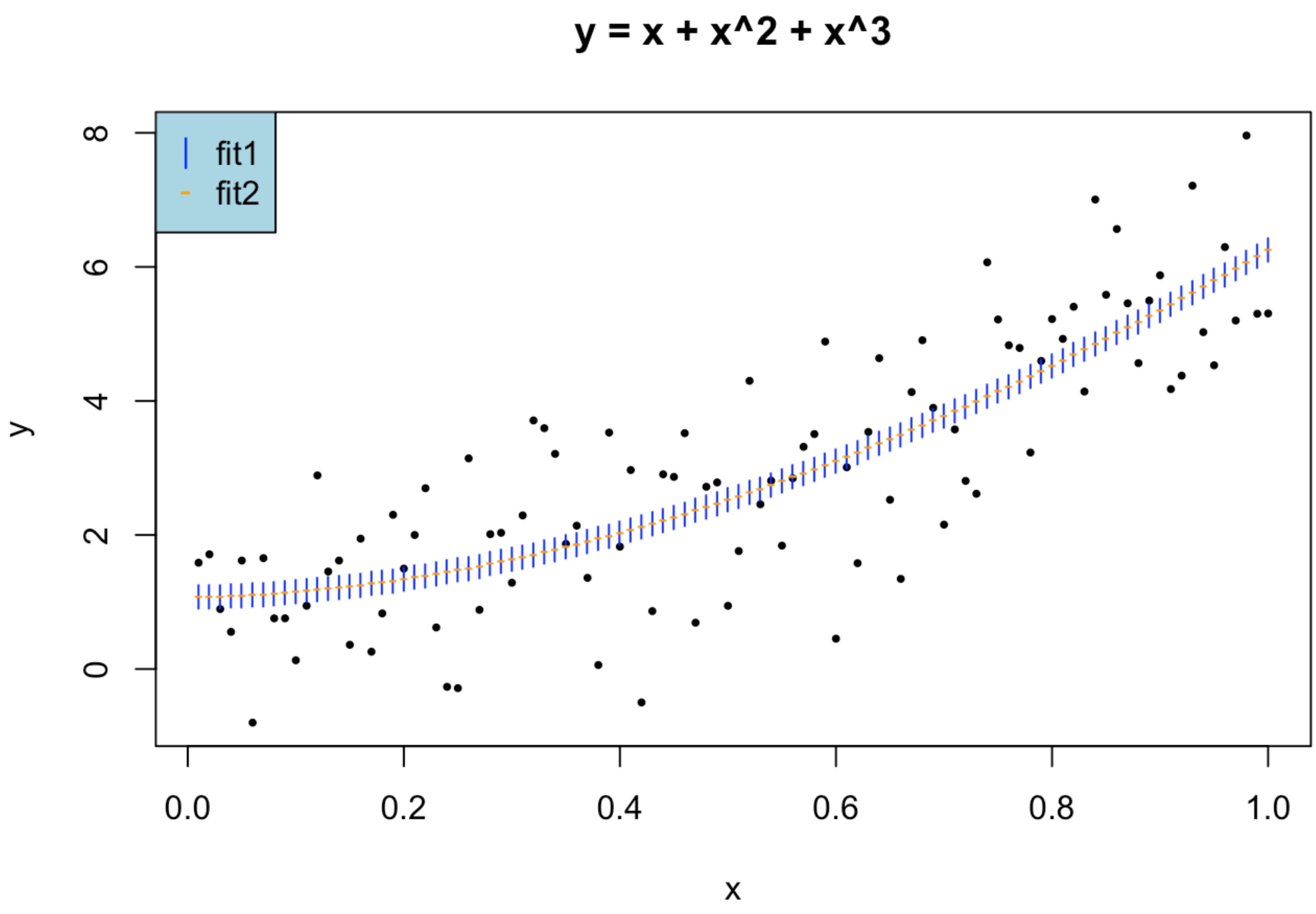
Because the p-values from F-tests on both models are smaller than 0.05, we can conclude that the fitted model is better than error only model at a significance level of 5%.

(iii)

```
plot(y ~ x, pch = 20, cex = 0.6,
    xlab = 'x',
    ylab = 'y',
    main = 'y = x + x^2 + x^3')

pred1 = predict(fit1)
pred2 = predict(fit2)
points(x, pred1, pch = '|', cex = 0.8, col = 'blue')
points(x, pred2, pch = '-', cex = 0.8, col = 'orange')

legend('topleft',
       legend = c('fit1','fit2'),
       col = c('blue','orange'),
       pch = c('|','-'),
       bg = 'lightblue')
```



(iv)

Based on the summary report from `fit2`, y_i is a second order polynomial in the predictor variable x_i

Although all p-values from summary of `fit1` are large, that does not mean all predictors have no prediction power on the response variable. A proper interpretation is that at the presence of all predictors in the current model, none of them contribute significantly to the prediction.

However, if we remove some predictors from the model, the p-value will change. See the change in p-value in the example below for removing the third order polynomial predictor from the model.

```
summary(lm(y~poly(x,2,raw=T)))
```

```
##
## Call:
## lm(formula = y ~ poly(x, 2, raw = T))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6508 -0.8398  0.2112  0.6546  2.1583
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.0357     0.3438   3.013  0.00330 **
## poly(x, 2, raw = T)1   0.7216     1.5711   0.459  0.64704
## poly(x, 2, raw = T)2   4.5422     1.5071   3.014  0.00329 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.123 on 97 degrees of freedom
## Multiple R-squared:  0.6682, Adjusted R-squared:  0.6613
## F-statistic: 97.65 on 2 and 97 DF,  p-value: < 2.2e-16
```

Question 4

(a)

$\frac{\partial SSE[\beta]}{\partial \beta} = \frac{\partial}{\partial \beta}(y - X\beta)^T(y - X\beta) = \frac{\partial}{\partial \beta}(y^T y - 2y^T X\beta + \beta^T\beta(X^T X)) = -2X^T y + 2\beta(XX^T)$

By setting $\frac{\partial SSE[\beta]}{\partial \beta}$ to zero, we obtain $\hat{\beta} = (X^T X)^{-1}X^T y$

(b)

$\frac{\partial}{\partial \beta}\Lambda = -2X^T y + 2\beta(XX^T) - \lambda^T C$

By setting $\frac{\partial}{\partial \beta} = 0$, we can obtain $\hat{\beta}_c = \frac{1}{2}(\lambda^T C)(XX^T)^{-1} + X^T y(XX^T)^{-1}$

Since we also have the constraint $C\beta - d = 0$, therefore by plugging in the previous result, we have

$$d - [\tfrac{1}{2}(\lambda^T C)(XX^T)^{-1} + X^T y(XX^T)^{-1}] \cdot C = 0$$

$$d - C \cdot (X^T y(XX^T)^{-1}) = C \cdot [\tfrac{1}{2}(\lambda^T C)(XX^T)^{-1}]$$

$$\tfrac{1}{2}(\lambda^T C)(XX^T)^{-1} = \frac{d - C\cdot(X^T y(XX^T)^{-1})}{C}$$

Taking the result back into the function for $\hat{\beta}_c$, we now have

$\hat{\beta}_c = \frac{d - C\cdot(X^T y(XX^T)^{-1})}{C} + X^T y(XX^T)^{-1}$

$\because \hat{\beta}_u = X^T y(XX^T)^{-1}, C^{-1} = (X^T X)^{-1}C^T[C(X^T X)^{-1}C^T]^{-1}$

$\therefore \hat{\beta}_c = \frac{d - C\hat{\beta}_u}{C} + \hat{\beta}_u = \hat{\beta}_u + C^{-1}(d - C\hat{\beta}_u) = \hat{\beta}_u + (X^T X)^{-1}C^T[C(X^T X)^{-1}C^T]^{-1}(d - C\hat{\beta}_u)$